Exercises

Labs

There are 6 exercises in this lab, of which the last exercise is "if time permits". Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

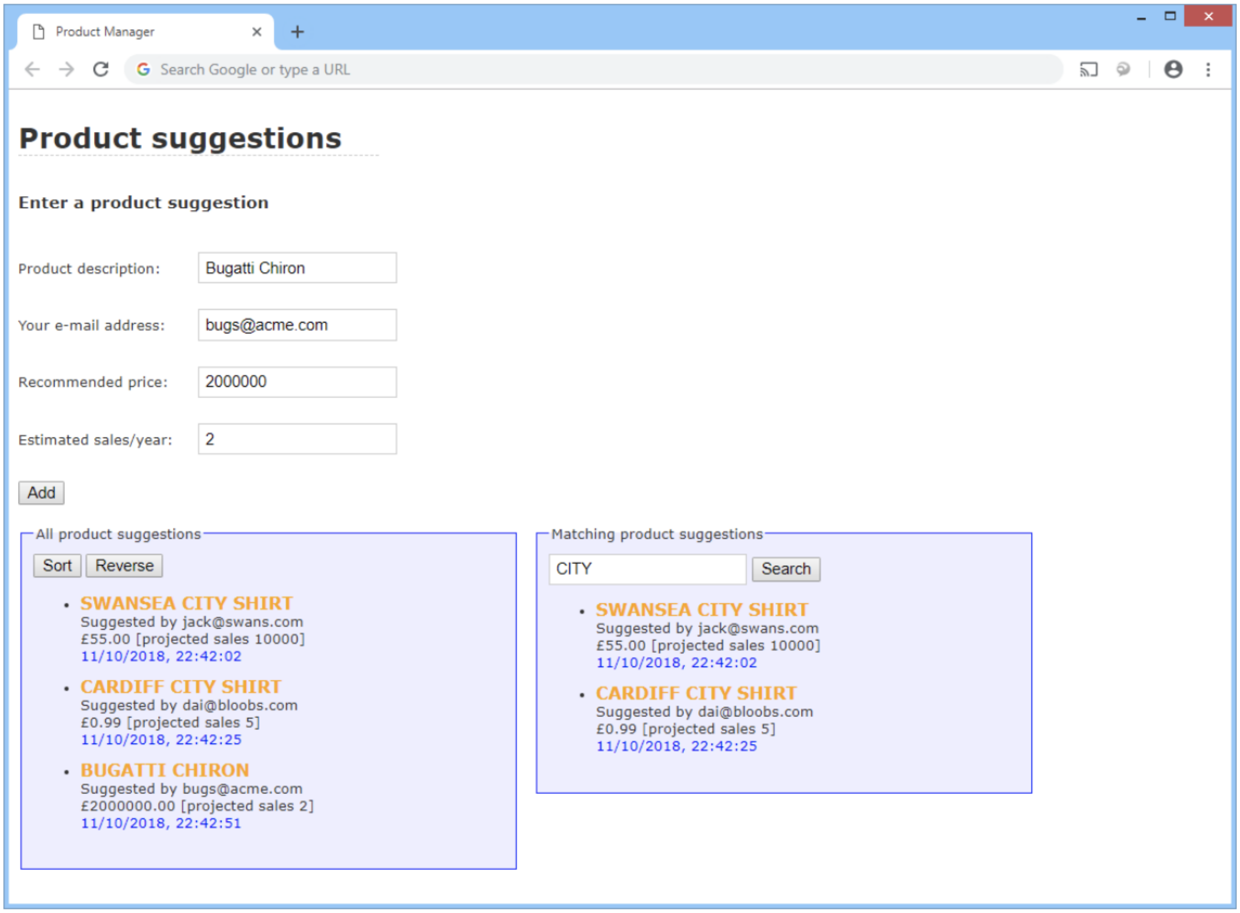
1. Getting started
2. Declaring variables using let and const
3. Using string interpolation and multi-line strings
4. Using arrow functions
5. Using default parameters
6. (Extra credit) Additional suggestions

Exercise 1: Getting started

In your file finder, go to the student folder for this lab. Note there are 2 script files:

- es6scripts/script.js - You will add all your ES6++ code in this file
- es5scripts/script.js - This file will be generated by transpiling the above file

Open index.html in a browser. The web page lets the user suggest a product that an online retailer might consider selling. The user enters their email address, along with a recommended price and an estimated sales/year. The web page also lets the user sort the items alphabetically, reverse the display order, and find products that match a specified search string.



Product suggestions

Enter a product suggestion

Product description: Bugatti Chiron

Your e-mail address: bugs@acme.com

Recommended price: 2000000

Estimated sales/year: 2

All product suggestions

- SWANSEA CITY SHIRT
  Suggested by jack@swans.com
  £55.00 [projected sales 10000]
  11/10/2018, 22:42:02
- CARDIFF CITY SHIRT
  Suggested by dai@bloobs.com
  £0.99 [projected sales 5]
  11/10/2018, 22:42:25
- BUGATTI CHIRON
  Suggested by bugs@acme.com
  £2000000.00 [projected sales 2]
  11/10/2018, 22:42:51

Matching product suggestions

CITY

- SWANSEA CITY SHIRT
  Suggested by jack@swans.com
  £55.00 [projected sales 10000]
  11/10/2018, 22:42:02
- CARDIFF CITY SHIRT
  Suggested by dai@bloobs.com
  £0.99 [projected sales 5]
  11/10/2018, 22:42:25

In this lab you'll refactor the script file to use ES6++ techniques (and then transpile it into ES5, for browser compatibility).

We've provided a Gulp file that automatically transpiles ES6++ files in any folder named es6scripts into an equivalent ES5 file in a folder named es5scripts. Open a terminal window in the student folder and run Gulp as follows. It will automatically transpile your `es6scripts/script.js` file into `es5scripts/script.js`:

```
npx gulp
```

Exercise 2: Declaring variables using let and const

Modify the code in es6scripts/script.js so that all the variables are declared using let/const rather than var. Generally, you'll always use let/const in ES6++, to avoid the "scope hoisting" problems of var variables.

Save es6scripts/script.js. Verify Gulp regenerates es5scripts/script.js, and take a look at the ES5 code. Then refresh the web page in the browser and verify it still works.

Exercise 3: Using string interpolation and multi-line strings

Modify es6scripts/script.js so that it uses string interpolation and multi-line strings to generate the various HTML strings. When you're done, save the file and verify it transpiles OK. Then refresh the web page and make sure it all still works.

Exercise 4: Using arrow functions

In es6scripts/script.js, take a look at the following code near the start of the file:

```
document.addEventListener('DOMContentLoaded', function () {
 document.getElementById('add').addEventListener('click', doAdd);
 document.getElementById('sort').addEventListener('click', doSort);
 document.getElementById('reverse').addEventListener('click', doReverse);
 document.getElementById('search').addEventListener('click', doSearch);
});
```

Change the anonymous callback function into an arrow function.

Also note the click handlers for the Sort and Reverse buttons – these statements specify the doSort and doReverse functions as event handlers. Modify the code to implement the logic of doSort and doReverse as arrow functions, rather than as separate functions. You can then delete the original doSort and doReverse functions themselves.

Test the code transpiles successfully and the web page still behaves correctly.

Exercise 5: Using default parameters

In es6scripts/script.js, take a look at the displayProducts function. The 2nd parameter is the id of the in which to display the products. In most cases, the products are displayed in the 'allProductsList' element, so set this as a default value for the parameter.

Then find all calls to the displayProducts function in the code. Wherever it passes in 'allProductsList' as the 2nd parameter, remove this parameter from the call and rely on the default value for this parameter instead.

Test the code transpiles successfully and the web page still behaves correctly

Exercise 6 (Extra credit): Additional suggestions

Take a look through the chapter and see if there are any opportunities to try out any additional
modern JavaScript features in your code.